



# Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique

Guillaume Bonfante, Bruno Guillaume, Mathieu Morey, Guy Perrier

## ► To cite this version:

Guillaume Bonfante, Bruno Guillaume, Mathieu Morey, Guy Perrier. Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique. Traitement Automatique des Langues Naturelles - TALN 2010, Jul 2010, Montréal, Canada. inria-00526437

**HAL Id: inria-00526437**

**<https://inria.hal.science/inria-00526437>**

Submitted on 14 Oct 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique

Guillaume Bonfante, Bruno Guillaume, Mathieu Morey, Guy Perrier  
INRIA Nancy-Grand Est - LORIA - Nancy-Université

**Résumé.** Nous définissons le  $\beta$ -calcul, un calcul de réécriture de graphes, que nous proposons d'utiliser pour étudier les liens entre différentes représentations linguistiques. Nous montrons comment transformer une analyse syntaxique en une représentation sémantique par la composition de deux jeux de règles de  $\beta$ -calcul. Le premier souligne l'importance de certaines informations syntaxiques pour le calcul de la sémantique et explicite le lien entre syntaxe et sémantique sous-spécifiée. Le second décompose la recherche de modèles pour les représentations sémantiques sous-spécifiées.

**Abstract.** We define the  $\beta$ -calculus, a graph-rewriting calculus, which we propose to use to study the links between different linguistic representations. We show how to transform a syntactic analysis into a semantic analysis via the composition of two sets of  $\beta$ -calculus rules. The first one underlines the importance of some syntactic information to compute the semantics and clearly expresses the link between syntax and underspecified semantics. The second one breaks up the search for models of underspecified semantic representations.

**Mots-clés :** Dépendances, réécriture de graphes, interface syntaxe-sémantique, DMRS.

**Keywords:** Dependencies, graph rewriting, syntax-semantics interface, DMRS.

## Introduction

Les analyseurs syntaxiques produisent des sorties dans des formats les plus divers. En outre, ils n'ont pas forcément en vue la production d'une représentation sémantique complète des énoncés et ne fournissent pas toujours les informations qui y seraient nécessaires. D'un autre côté, il existe différents formalismes de représentation sémantique liés ou non à des formalismes de représentation syntaxique.

D'un point de vue théorique comme d'un point de vue pratique, il apparaît utile de disposer d'un cadre formel commun qui permette d'étudier la transformation d'une représentation dans un format A en une représentation dans un format B, les formats A et B ayant été développés de façon totalement indépendante. Posé de façon totalement générale, le problème est insoluble. Pour cette étude, nous nous restreindrons à des formats en dépendances, tant pour la syntaxe que pour la sémantique.

En effet, de nombreux analyseurs syntaxiques produisent des arbres de dépendances, soit de façon directe ou par transformation d'arbres syntagmatiques. La récente campagne PASSAGE d'évaluation des analyseurs syntaxiques du français<sup>1</sup> en est une bonne illustration. Pour cette campagne, a été défini un format

---

1. <http://atoll.inria.fr/passage/>

de sortie qui prend en compte à la fois la structure en constituants des phrases et la structure en dépendances, appelées ici relations. Dans PASSAGE, on peut choisir indifféremment de faire porter les relations sur les mots ou sur des groupes syntaxiques (qui ne sont pas forcément des syntagmes complets), mais la présence d’une tête dans chaque groupe syntaxique permet de faire systématiquement porter les relations sur les mots. Chaque phrase est alors annotée par un graphe de dépendances entre mots, rien n’interdisant des cycles dans les dépendances comme pour la phrase « *Jean me propose un livre facile à lire* », où le cycle de dépendances est localisé dans le segment « *livre facile à lire* ».

Pour ce qui est de la sémantique, les formats en dépendances sont la règle pour les grammaires de dépendance (voir la Théorie Sens-Texte par exemple (TST) (Mel’čuk, 1988)). Mais c’est aussi le cas pour des représentations produites à partir de grammaires syntagmatiques. Ainsi, à partir de HPSG, a été conçu un formalisme de représentation sémantique sous-spécifiée la MRS (*Minimal Recursion Semantics*) (Copestake *et al.*, 2005). De ce formalisme, a été dérivé un formalisme équivalent plus compact et plus lisible, où les structures sémantiques sont des graphes de dépendances ; c’est la *Dependency Minimal Recursion Semantics* (DMRS) (Copestake, 2009).

De nombreux travaux en linguistique utilisent des graphes en syntaxe et en sémantique. Notre travail se place dans ce cadre mais se concentre sur la formalisation des processus de calcul entre ces structures.

Pour transformer les graphes de dépendances syntaxiques en graphes de dépendances sémantiques, nous proposons d’utiliser une forme particulière de *réécriture de graphes*, que nous avons appelé  $\beta$ -calcul. Un système du  $\beta$ -calcul est un ensemble de *règles* qui permettent de transformer un graphe d’un certain formalisme dans un graphe dans un autre formalisme. L’intérêt d’utiliser le paradigme de la réécriture de graphes est d’une part de pouvoir exprimer des principes linguistiques de façon simple et lisible, chaque règle traduisant un principe linguistique. D’autre part, cela permet d’exploiter la souplesse inhérente à la réécriture de graphes. On peut ainsi enrichir une transformation d’un format A en un format B de façon complètement monotone par simple ajout de règles. On peut aussi composer deux transformations d’un format A à un format B et d’un format B à un format C par union des deux ensembles de règles, si on suppose que le second ne peut pas interférer avec le premier, en ce sens qu’une règle du second ne s’applique jamais avant une règle du premier.

Pour illustrer notre propos, nous définissons un ensemble de  $\beta$ -règles qui permettent de transformer des structures de dépendances syntaxiques au format PASSAGE en structures de dépendances sémantiques au format DMRS. Nous avons choisi ces deux formats, car l’information qu’ils représentent, si elle n’est pas complète, fait consensus et ils s’appuient tous les deux sur une expérimentation à large échelle : la campagne d’évaluation PASSAGE d’un côté et le projet DELPH-IN<sup>2</sup> de l’autre.

Dans la section 1, nous montrerons que l’information donnée par les structures de dépendances PASSAGE n’est pas suffisante pour produire des structures de dépendances sémantiques DMRS. Nous décrirons précisément les relations qu’il faut ajouter, les étiquettes des relations existantes qu’il faut raffiner et l’information lexicale qu’il faut mettre sur les mots pour obtenir ce que nous appellerons un format enrichi DSC (pour Dépendances Syntaxiques Complètes). Nous décrirons aussi le format DMRS que nous visons.

Contrairement à la réécriture de termes ou de mots, il y a plusieurs définitions possibles de la réécriture de graphes (Rozenberg, 1997). Chacune d’entre elles induit des propriétés calculatoires spécifiques, en particulier une sémantique opérationnelle caractéristique. De ce fait, le choix de la définition est guidé par le domaine d’application. Dans notre étude, un nouveau calcul est nécessaire : le  $\beta$ -calcul, présenté à la

---

2. <http://www.delph-in.net/>

section 2. Notre approche est compatible avec des approches antérieures, comme celle de (Bédaride & Gardent, 2008) qui traite des inférences textuelles.

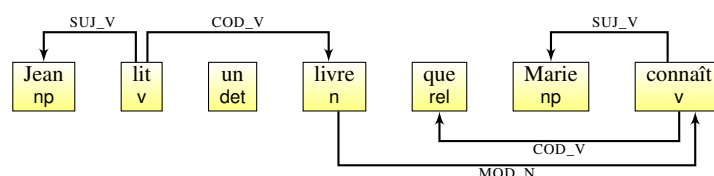
Dans la section 3, nous décrivons un ensemble  $R_1$  de  $\beta$ -règles qui permettent de transformer les structures de dépendances syntaxiques au format DSC en structures de dépendances sémantiques DMRS. Ainsi, sera établi de façon formelle la proximité du format PASSAGE par rapport au format DMRS. Plus même, nous pointerons exactement quelles sont les informations manquantes du format PASSAGE pour produire des représentations DMRS.

Dans la section 4, nous illustrerons une autre facette de la souplesse du  $\beta$ -calcul, son aptitude à composer des transformations. Nous décrivons un ensemble  $R_2$  de  $\beta$ -règles qui permettent d'obtenir, à partir des représentations sémantiques sous-spécifiées de la DMRS, des modèles de ces représentations correspondant à différentes formes logiques, où les portées des quantificateurs et des adverbes ont été complètement fixées. Nous appellerons le format dans lequel ces modèles sont définis le format DMRSF (F pour final).

## 1 Enrichir le format PASSAGE pour produire des structures DMRS

**Le format PASSAGE** Le format PASSAGE, si nous ignorons le volet des groupes syntaxiques, consiste avant tout en un jeu de 14 étiquettes de relations. Chacune représente une fonction syntaxique avec éventuellement en plus la catégorie grammaticale du gouverneur de la relation. Par exemple, on distingue la relation *modifieur de nom* (MOD\_N) de celle de *modifieur de verbe* (MOD\_V).

L'annotation en dépendances syntaxiques selon PASSAGE de la phrase « Jean lit un livre que Marie connaît » est donnée par la figure ci-dessous<sup>3</sup>.



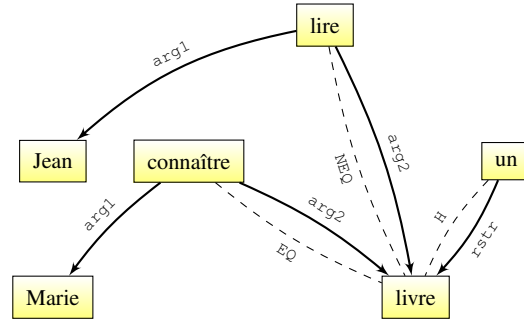
**Le format DMRS** Une structure en DMRS est un graphe dont les nœuds sont des prédicats et les arcs des relations entre ces prédicats. Ces arcs sont étiquetés par deux types d'informations :

- Si l'arc représente une relation prédicat-argument, une étiquette précise la nature de cette relation : `rstr` pour la restriction d'un quantificateur, `arg1`, `arg2`, `arg3` pour les relations entre un prédicat standard et ses arguments selon un ordre d'oblicité canonique.
- Des arcs étiquetés par `EQ` (resp. `NEQ`) indiquent que les deux prédicats sont (resp. ne sont pas) associés dans une même conjonction. L'étiquette `H` indique une relation de portée entre un prédicat et un élément de cette portée : la restriction ou le corps pour un quantificateur par exemple.

Reprenons l'exemple de la phrase « Jean lit un livre que Marie connaît ». Une représentation de sa sémantique en DMRS est présentée dans la figure ci-dessous, si l'on considère une lecture restrictive de la relative.

3. Pour se conformer à l'usage en grammaires de dépendances, le sens des dépendances est inversé par rapport au format officiel PASSAGE. Le déterminant « un » reste isolé car PASSAGE ignore les relations déterminant-nom.

L'arc étiqueté `rstr` indique que le quantificateur « *un* » a comme restriction le prédicat « *livre* » et la relation étiquetée `EQ` entre « *connaître* » et « *livre* » signifie que les deux prédicats doivent être liés dans une même conjonction, donc « *connaître* » est aussi dans la restriction de « *un* ». L'autre arc étiqueté `arg2` entre « *connaître* » et « *livre* » signifie que le prédicat « *connaître* » a comme argument en position 2 (l'objet direct du verbe dans sa forme canonique), un élément caractéristique du prédicat « *livre* ». La différence pour le prédicat « *lire* » vis-à-vis du prédicat « *connaître* » est qu'il est relié par un arc étiqueté `NEQ` à « *livre* », ce qui signifie que les deux prédicats ne sont pas associés dans la même conjonction, et donc que « *lire* » n'est pas dans la restriction de « *un* ».



**Le format DSC** Un examen informel d'exemples de phrases annotées en dépendances syntaxiques `PASSAGE` et en dépendances sémantiques `DMRS` montre que les deux représentations sont relativement proches mais qu'il manque certaines informations syntaxiques dans les dépendances `PASSAGE` pour obtenir les dépendances `DMRS`. Nous proposons ci-dessous un certain nombre d'enrichissements du format `PASSAGE` de départ pour produire le format `DSC` (dépendances syntaxiques complètes) et nous montrerons dans la section 3 un ensemble de  $\beta$ -règles permettant de passer de `DSC` à `DMRS` :

- Pour modéliser la quantification au niveau sémantique, il est nécessaire d'ajouter les relations `DET` déterminant-nom.
- Pour les propositions relatives, la relation anaphorique entre le pronom relatif et son antécédent est indispensable, même si on peut penser qu'elle relève de la résolution générale des anaphores, à la différence près qu'ici, elle est principalement déterminée par la syntaxe. Elle sera marquée par l'étiquette `ANT`.
- Les prépositions, quand elles introduisent des compléments requis, donnent des informations sur leur rôle sémantique, et lorsqu'elles introduisent des compléments adjoints, elles peuvent être considérées comme des prédicats à part entière. Nous différencierons les étiquettes `CPL_V` en `CPL_V_REQ` pour les compléments requis et `CPL_V_ADJ` pour les compléments adjoints.  
Par ailleurs, dans les deux cas, il est nécessaire de connaître la relation entre la préposition et la tête du groupe nominal ou de la proposition qui suit. Cette relation n'apparaît pas dans `PASSAGE` et elle doit être ajoutée. Comme (Candito *et al.*, 2009), on peut choisir `OBJ` pour l'étiqueter.
- Comme les analyseurs syntaxiques participant à `PASSAGE` travaillent avec des jeux d'étiquettes souvent incompatibles, `PASSAGE` a choisi un jeu contenant un minimum d'information mais pour la sémantique, il est nécessaire d'avoir une information plus précise. C'est par exemple le cas pour les relations `AUX` qui ont besoin d'être précisées en `TPS` ou `PASSIF`, selon que l'auxiliaire est un auxiliaire de temps ou du passif. On a aussi besoin, pour les relations `MOD_N` et `MOD_A` de traiter à part les compléments requis par les noms et les adjectifs, en spécifiant les étiquettes sous la forme par exemple de `CPL_N_REQ` et `CPL_A_REQ`.
- Enfin, il est nécessaire d'enrichir les nœuds des graphes avec des informations morphologiques, syntaxiques ou sémantiques, sous forme par exemple de structures de traits. Nous ne détaillerons pas ce point ici.

La motivation du format `DSC` ci-dessus est d'indiquer dans la même structure toute l'information nécessaire au transfert vers une structure sémantique avec un petit nombre de règles qui ne dépendent pas du

lexique. Dans notre cadre, les relations syntaxiques sont beaucoup plus riches que celles de Kahane (2003) par exemple où plus d'informations sont données dans l'interface plutôt que dans la structure syntaxique.

## 2 Formalisation des règles de calcul

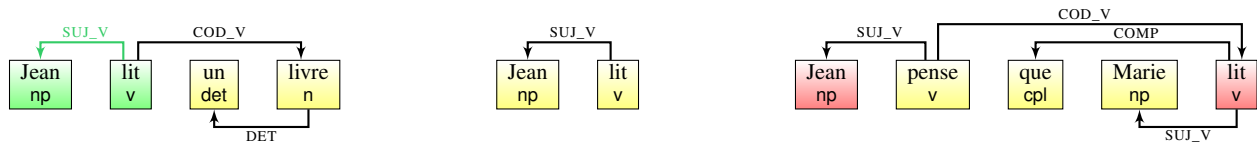
Le principe calculatoire de notre système est la réécriture de graphe, plus précisément la réécriture de  $\beta$ -structures. Un système de réécriture consiste en un ensemble fini de règles de la forme  $L \rightarrow_\phi R$  : si le motif  $L$  est reconnu dans une  $\beta$ -structure, il sera remplacé par  $R$ . On suppose fixé un ensemble fini d'étiquettes  $\mathcal{L}$ .

**Structure** Une  $\beta$ -structure est donnée par un  $k + 2$ -uplet  $(V, \ell, \rightarrow^1, \dots, \rightarrow^k)$  où  $V$  est un ensemble fini de sommets,  $\ell : V \rightarrow \mathcal{L}$  est une fonction d'étiquetage et pour tout  $i \leq k$ ,  $\rightarrow^i$  est une relation binaire (i.e.  $\rightarrow^i \subset V \times V$ ). En d'autres termes, on a un nombre fini de graphes sur le même ensemble de sommets. On utilisera la notation  $(V, \ell, \Rightarrow)$  comme une abréviation de  $(V, \ell, \rightarrow^1, \dots, \rightarrow^k)$ . Étant donnée une  $\beta$ -structure  $G$ , on notera respectivement  $V_G$ ,  $\ell_G$  et  $\Rightarrow_G$  les éléments du triplet correspondant.

Dans les exemples de cet article, les sommets correspondent aux entités lexicales et sont étiquetées par des structures de traits qui contiennent, entre autres, la catégorie syntaxique, le lemme ou la forme fléchiée par exemple. Les différentes relations sont représentées par des étiquettes (syntaxique ou sémantique) sur les arcs<sup>4</sup>.

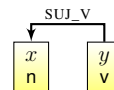
**Motif** Un motif  $L$  est une  $\beta$ -structure  $(V_L, \ell_L, \Rightarrow_L)$ . Reconnaître un motif  $L$  dans une  $\beta$ -structure (appelée *instance*)  $I = (V_I, \ell_I, \Rightarrow_I)$  consiste à construire un *morphisme*, c'est-à-dire une fonction  $\psi : V_L \rightarrow V_I$  telle que  $\forall x \in V : \ell_L(x) = \ell_I(\psi(x))$  et  $\forall x, y \in V : x \rightarrow_L^i y \Leftrightarrow \psi(x) \rightarrow_I^i \psi(y)$ . Nous supposons que ces morphismes sont toujours injectifs<sup>5</sup>.

En d'autres termes, étant donné un motif à  $n$  sommets, on doit distinguer dans l'instance le même nombre de sommets, chaque sommet identifié à la même étiquette que son antécédent et les sommets ont les mêmes relations entre eux.



Dans la figure ci-dessus, nous avons un tel monomorphisme entre le motif au centre et la phrase de gauche. Il est illustré par la mise en couleur verte.

Du fait de l'absence de liens entre « Jean » et « lit » dans la phrase de droite, le motif central n'est pas satisfait. Par la suite, pour que les règles soient suffisamment génériques, nous paramétrons les étiquettes. Le motif ci-contre recouvre ainsi tous les motifs pour lesquels  $x$  est un *nom* et  $y$  un *verbe*, les deux étant reliés par une relation  $SUI\_V$ .

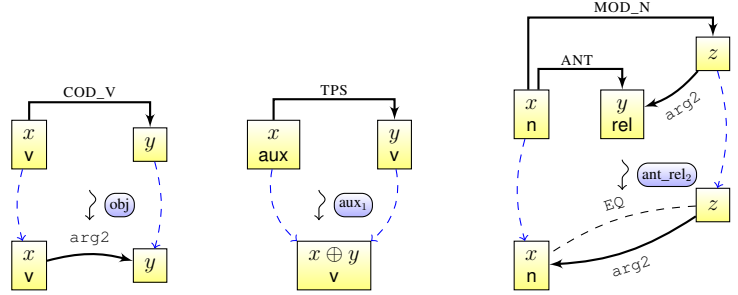


4. Il est en fait équivalent de considérer un ensemble de relations ou une relation étiquetée.

5. Techniquement,  $\psi$  est un monomorphisme plein.

**Réécriture** Une *règle* est un triplet  $L \rightarrow_\phi R$  où  $L$  et  $R$  sont des  $\beta$ -structures et  $\phi$  est une fonction de  $V_L$  dans  $V_R$ . La  $\beta$ -structure  $L$  est le motif à reconnaître, motif qui est remplacé par  $R$ . La fonction  $\phi$  décrit la manière de faire le recollement : les sommets de  $L$  sans image sont détruits, deux sommets ayant le même but sont fusionnés.

Dans les trois règles ci-contre, en haut, on trouve le motif, en bas son remplacement. Les flèches bleues pointillées décrivent la fonction  $\phi$ . En fait, les noms  $x$ ,  $y$  et  $z$  employés pour les étiquettes rendent compte de cette fonction  $\phi$ .  $x \oplus y$  dénote la fusion de deux nœuds  $x$  et  $y$ .

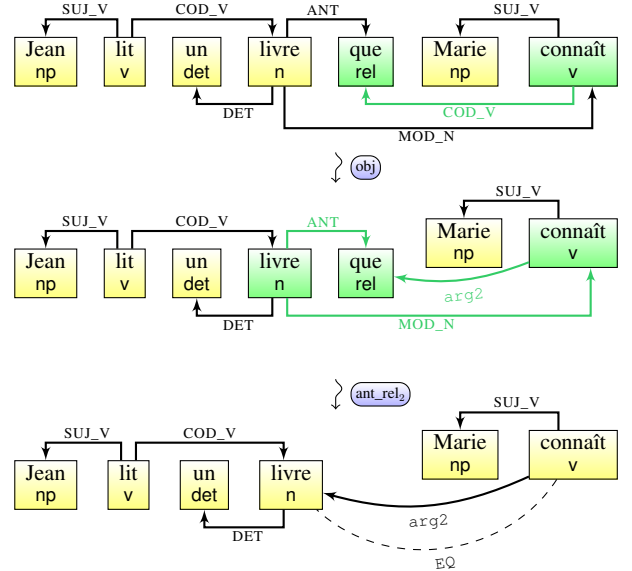


Dans la règle  $\text{aux}_1$ , les deux nœuds  $x$  et  $y$  sont fusionnés. La structure de traits associée à la fusion contient en priorité les traits spécifiés dans la règle (la catégorie syntaxique  $v$  dans l'exemple), et l'unification des traits des nœuds  $x$  et  $y$  pour les autres traits. Dans la règle  $\text{ant\_rel}_2$ , le sommet  $y$  est détruit.

L'application d'une règle  $L \rightarrow_\phi R$  sur une  $\beta$ -structure  $I$  avec le morphisme  $\psi : L \rightarrow I$  produit une nouvelle  $\beta$ -structure  $F$ . En notant le contexte  $C = V_I \setminus \psi(V_L)$ , on définit  $F = (C + V_R, \ell_{L|C} + \ell_R, \Rightarrow_F)$  telle que  $x \rightarrow_F^i y$  est vraie ssi l'une des conditions suivantes est vraie :

- (a)  $x, y \in V_R$  et  $x \rightarrow_R^i y$  (liens internes du motif de remplacement),
- (b)  $x, y \in C$  et  $x \rightarrow_I^i y$  (liens internes du contexte),
- (c)  $x = \phi(x')$  avec  $x' \in V_L$ ,  $y \in C$  et  $\psi(x') \rightarrow_I^i y$  (lien du motif qui pointent vers un mot du contexte),
- (d)  $x \in C$ ,  $y = \phi(y')$  avec  $y' \in V_L$  et  $x \rightarrow_I^i \psi(y')$  (lien du contexte qui pointent vers un mot du motif).

Un exemple d'applications des règles de réécriture est donnée dans la figure ci-contre.



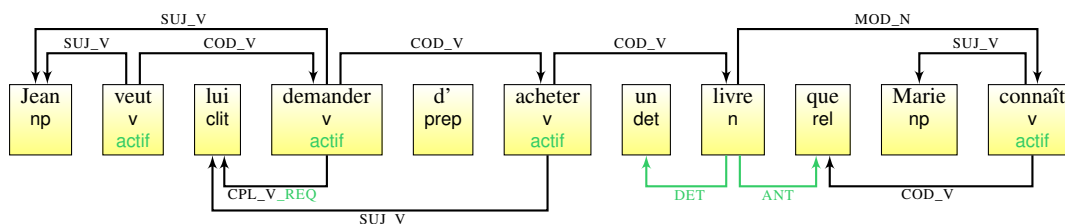
### 3 De la syntaxe complète à la sémantique sous-spécifiée

Nous pouvons maintenant montrer concrètement comment des règles de  $\beta$ -calcul permettent de calculer une représentation sémantique sous-spécifiée au format DMRS à partir de dépendances syntaxiques complètes au format DSC. Fournir une procédure de calcul de la sémantique à partir de la syntaxe n'est toutefois pas notre unique but ici : l'examen de la forme des  $\beta$ -règles utilisées permet de répondre à deux questions.

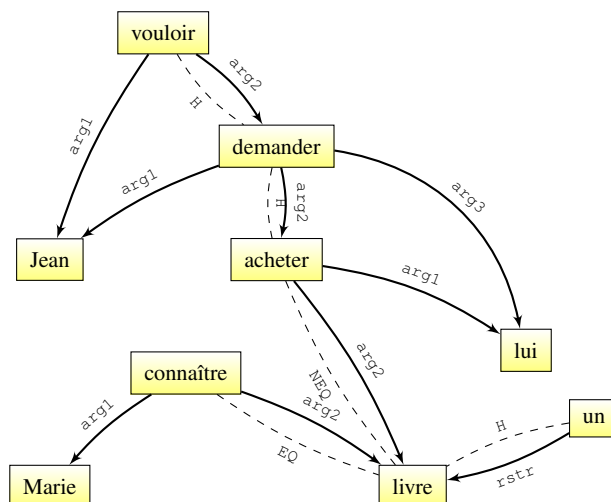
D'une part, sur quelles informations lexicales et syntaxiques les règles s'appuient-elles ? Cela nous renseigne essentiellement sur ce que le format de sortie (ici la DMRS) attend du format d'entrée (ici DSC). D'autre part, les règles sont-elles complexes ? De façon caricaturale : les règles font-elles de simples

changements d'étiquettes, ou modifient-elles lourdement la structure du graphe ? Cela nous renseigne sur l'écart entre les deux formats considérés : plus les règles sont complexes en moyenne, plus l'information décrite par les formats est de nature différente.

À titre d'exemple, considérons la phrase « *Jean veut lui demander d'acheter un livre que Marie connaît* ».



L'analyse DSC de cette phrase montre qu'elle combine un verbe à contrôle par le sujet « *veut* », un verbe à contrôle par le complément d'attribution « *demander* » et une subordonnée relative « *que Marie connaît* ». En vert, on distingue les ajouts faits à PASSAGE : « *lui* » est un complément requis (et qui n'est ni sujet ni objet direct) du verbe, « *un* » est le déterminant de « *livre* », l'antécédent du pronom relatif « *que* » est « *livre* », et les verbes sont à la voix active. Comme nous l'avons expliqué en section 1, ces informations sont nécessaires pour le calcul de la représentation sémantique en DMRS. En particulier, connaître la voix (active, passive ou moyenne) du verbe permet de lui attribuer les bons arguments sémantiques.

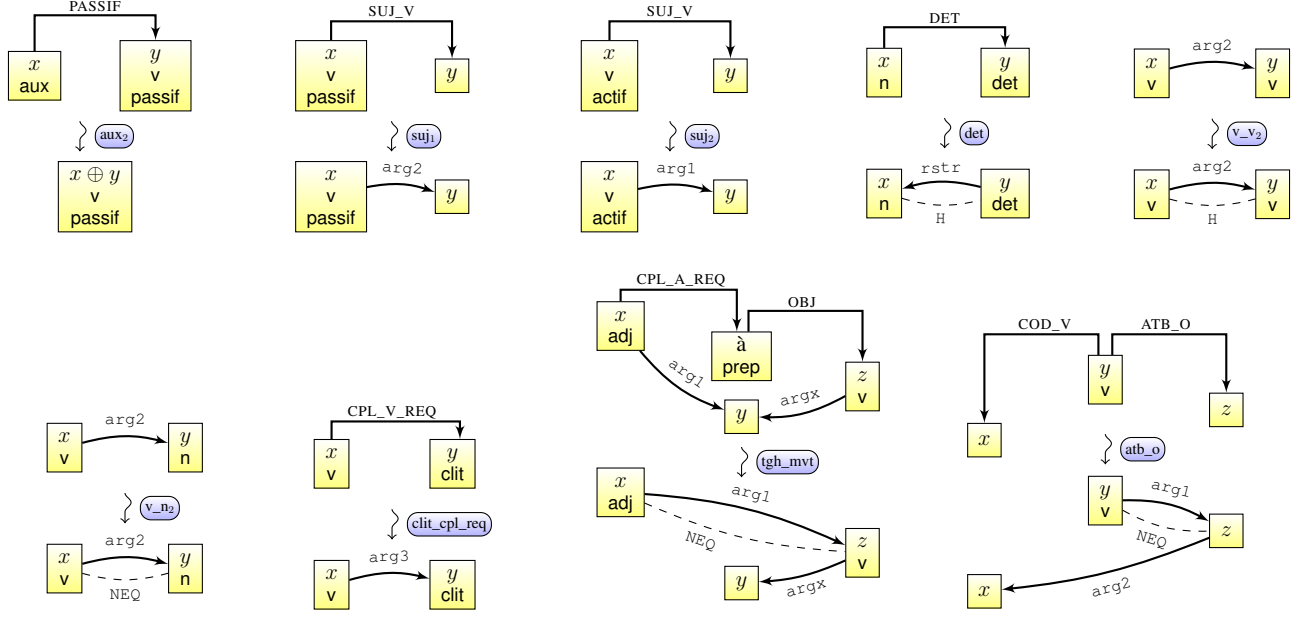


Pour notre exemple, le calcul de la représentation DMRS utilise sept  $\beta$ -règles : *obj* et *ant\_rel<sub>2</sub>* ont été présentées en partie 2, *det*, *subj<sub>2</sub>*, *v<sub>v2</sub>*, *v<sub>n2</sub>* et *clit\_cpl\_req* font partie du fragment donné ci-dessous. Le jeu complet DSC-DMRS comporte une trentaine de règles.

Ces règles correspondent toutes à des principes linguistiques : par exemple, *subj<sub>1</sub>* traduit le fait que l'*arg2* d'un verbe à la voix passive est fourni par son sujet, *subj<sub>2</sub>* traduit le fait que l'*arg1* d'un verbe à la voix active est fourni par son sujet, *det* traduit le fait que le nom est dans la restriction (*rstr* et *H*) du quantificateur introduit par son déterminant. . .

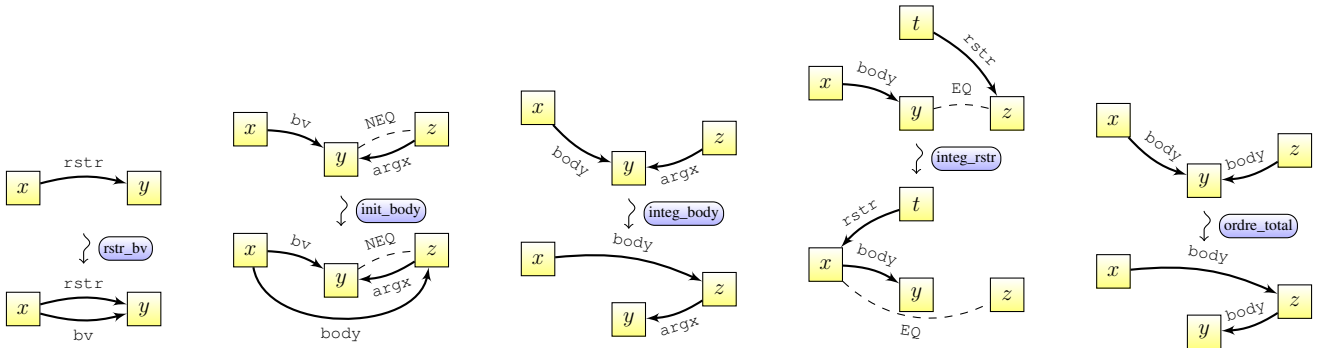
La simplicité de ces règles confirme une intuition : pour l'essentiel, les dépendances syntaxiques complètes et les dépendances sémantiques DMRS sont très proches. Quelques phénomènes linguistiques laissent cependant un écart plus grand entre syntaxe complète et sémantique. C'est le cas par exemple du *tough movement* et de l'attribut de l'objet. La règle *tgh\_mvt* s'applique à des expressions comme « *un livre facile à lire* » : ce n'est pas le livre lui-même qui est facile, c'est sa lecture. La règle de réécriture l'exprime en produisant une structure qui correspond (approximativement) à la paraphrase « *un livre qui est facile à lire* ». La règle *ath\_o* s'applique à des expressions comme « *Jean trouve ce livre difficile* » : alors que dans l'analyse syntaxique, l'objet (« *livre* ») et son attribut (« *difficile* ») ne sont pas liés, dans l'analyse sémantique l'attribut de l'objet a celui-ci pour argument. La lecture obtenue se rapproche de la paraphrase « *Jean trouve que ce livre est difficile* ».





## 4 De la sémantique sous-spécifiée à ses modèles

Le  $\beta$ -calcul permet également de calculer des modèles DMRSF équivalents aux formules logiques qui correspondent aux différentes lectures de la phrase. Dans un modèle DMRSF la portée de chaque quantificateur, verbe ou adverbe modal est fixée. Cette portée se décompose en une restriction, notée *rstr*, et un corps, noté *body*. On présente ici une partie du jeu de règles qui génère les modèles DMRSF pour une structure DMRS donnée. En composant ce jeu de règles avec celui présenté à la section 3, on peut calculer les modèles DMRSF à partir d'une analyse syntaxique complète DSC.



Les règles correspondent aux phases successives du calcul de la portée des quantificateurs :

**rstr\_bv** : D'abord on introduit un arc *bv* (pour Bound Variable) entre le quantificateur et le nom commun porteur de sa variable caractéristique. Cet arc restera fixe dans le calcul des modèles, contrairement à l'arc *rstr* qui bougera au fur et à mesure du calcul de la restriction du quantificateur.

**init\_body** : On initialise ensuite le corps (noté *body*) du quantificateur : tous les prédicats qui sont dans sa portée (ils ont pour argument la variable liée par le quantificateur) mais pas dans sa restriction (le lien est de type *NEQ*) sont forcément dans son corps.

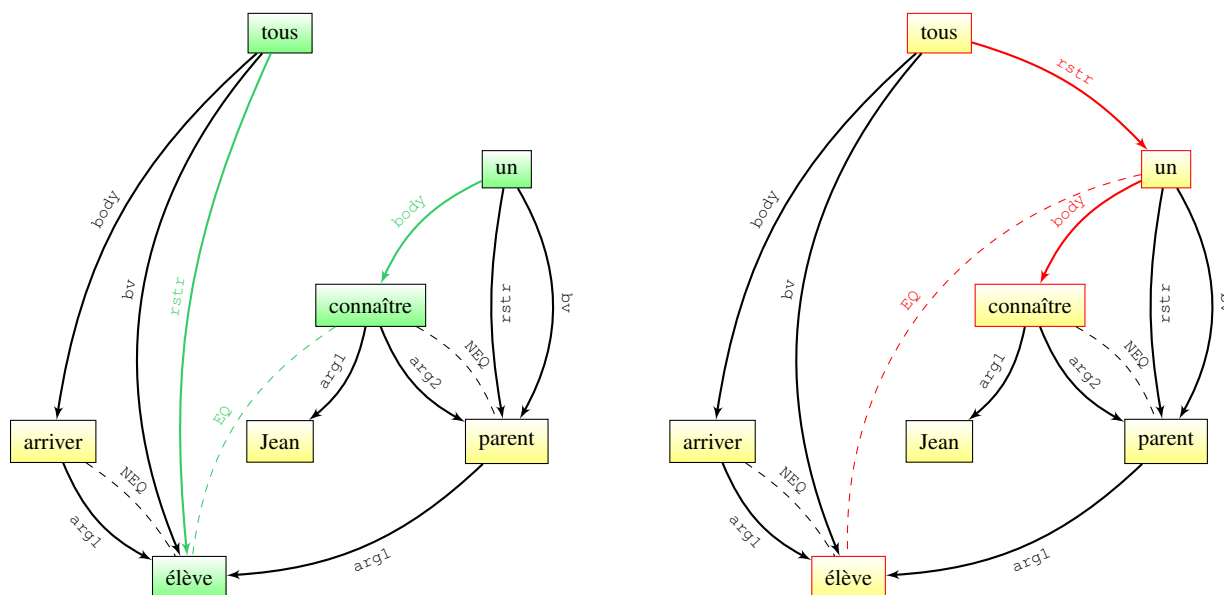
**integ\_body** : Si un prédicat (verbe, adverbe, adjectif, nom relationnel...) a un argument qui se trouve dans le corps de la portée d'un quantificateur, ce prédicat fait lui aussi partie du corps de la portée.

**integ\_rstr** : La restriction d'un quantificateur peut contenir entièrement des quantificateurs de plus petite portée.

**ordre\_total** : S'il y a ambiguïté de portée entre deux quantificateurs, on génère les deux lectures possibles. Étant donnés deux quantificateurs A et B, une application de la règle donnera portée large à A sur B et une autre application portée large à B sur A.

Considérons la phrase « *Tous les élèves dont Jean connaît un parent sont arrivés* ». Pour obtenir le modèle DMRSF équivalent à la formule  $Tous[x, \text{eleve}(x) \wedge Un[y, \text{parent}(y, x), \text{connaître}(j, y)], \text{arriver}(x)]$  en théorie des quantificateurs généralisées, on applique à la représentation DMRS de la phrase les  $\beta$ -règles suivantes :

1. *rstr\_bv* fixe le lien *bv* entre le quantificateur « *tous* » et « *élèves* » (le nom commun qui porte sa variable caractéristique),
2. *rstr\_bv* fixe le lien *bv* entre le quantificateur « *un* » et « *parent* »,
3. *init\_body* amorce le corps de la portée de « *tous* » ; celui-ci contient « *arriver* »,
4. *init\_body* amorce le corps de la portée de « *un* » ; celui-ci contient « *connaître* » et donc son argument « *Jean* »,
5. *integ\_rstr* intègre la sémantique de toute la subordonnée relative dans la restriction de « *tous* » ; « *un* » et toute sa portée font donc partie de la restriction de « *tous* ». Cette application de règle est représentée ci-dessous.



## Conclusion

Nous avons utilisé le  $\beta$ -calcul, un calcul de réécriture de graphes, pour examiner le lien entre dépendances syntaxiques et sémantiques. En particulier, nous avons montré qu'en augmentant le format en dépendances syntaxiques PASSAGE, on pouvait générer simplement une représentation sémantique sous-spécifiée en DMRS et, de là, des représentations logiques correspondant aux lectures d'un énoncé. Le  $\beta$ -calcul fournit un cadre commun à toutes les phases du calcul, ce qui contribue à éclaircir les liens entre les différents formats de représentation. Pour éviter la surgénération, nous projetons d'affiner notre jeu de  $\beta$ -règles par une expérimentation à large échelle.

Au même titre que la construction d'un grand programme, construire un grand jeu de règles est un problème difficile. Parmi les soucis, notons ceux concernant la terminaison, la modularité, la cohérence des règles entre elles, leur compilation, etc. Notre approche formelle nous permet de réemployer les techniques traditionnelles de construction d'invariants, en d'autres termes, de pouvoir déterminer automatiquement certaines des propriétés de nos systèmes de règles. Nous travaillons activement à ce sujet.

## Références

- BÉDARIDE P. & GARDENT C. (2008). Réécriture et détection d'implication textuelle. In *Traitement Automatique des Langues Naturelles - TALN 2008*, p. 19–28, Avignon France.
- CANDITO M.-H., CRABBÉ B., DENIS P. & GUÉRIN F. (2009). Analyse syntaxique statistique du français : des constituants aux dépendances. In *TALN 2009*, Senlis, France.
- COPESTAKE A. (2009). *Invited Talk* : Slacker semantics : Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, p. 1–9, Athens, Greece : Association for Computational Linguistics.
- COPESTAKE A., FLICKINGER D., POLLARD C. & SAG I. (2005). Minimal Recursion Semantics - an Introduction. *Research on Language and Computation*, **3**, 281–332.
- KAHANE S. (2003). On the status of deep syntactic structure. In *Proceedings of the First International Conference on Meaning-Text Theory (MTT'03)*, p. 229–238, Paris.
- MEL'ČUK I. (1988). *Dependency Syntax : Theory and Practice*. Albany : State Univ. of New York Press.
- G. ROZENBERG, Ed. (1997). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1 : Foundations*. World Scientific.